

ESTIMACIÓN DE LA REGRESIÓN LASSO

Miguel Coto-García¹ & Natalia Díaz-Ramírez¹

¹Estudiante de Maestría Profesional en Estadística, Universidad de Costa Rica, San Pedro, Costa Rica

1. INTRODUCCIÓN

La selección de variables es un procedimiento utilizado para incluir las variables con mayor significancia en el análisis, entre los métodos más utilizados se encuentran los métodos de eliminación por stepwise que sólo mejora la exactitud de las predicciones en ciertos casos, como cuando hay variables con un efecto importante en la variable independiente.

En cualquier modelo de regresión, como no se saben los parámetros verdaderos, β , se tienen que estimar a partir de la muestra. En el enfoque de Mínimos cuadrados ordinarios (el más usual), los parámetros se estiman como $\hat{\beta}$ de tal manera que la suma de cuadrados residual sea lo más pequeña posible. En otras palabras, se minimiza la siguiente función de pérdida y mide la exactitud de las estimaciones:

$$L_{OLS}(\hat{\beta}) = \sum_{i=1}^n (y_i - x_i' \hat{\beta})^2 = \|y - X\hat{\beta}\|^2$$

Por otro lado, en estadística, hay dos características críticas de los estimadores que deben considerarse: el sesgo y la varianza. El sesgo es la diferencia entre el parámetro de población real y el estimador esperado:

$$Bias(\hat{\beta}_{OLS}) = E(\hat{\beta}_{OLS}) - \beta.$$

La variación, por otro lado, mide la propagación, o incertidumbre, en estas estimaciones. Es dado por

$$Var(\hat{\beta}_{OLS}) = \sigma^2 (X'X)^{-1},$$

donde la varianza de error desconocida σ^2 se puede estimar a partir de los residuos como

$$\hat{\sigma}^2 = \frac{e'e}{n-m},$$

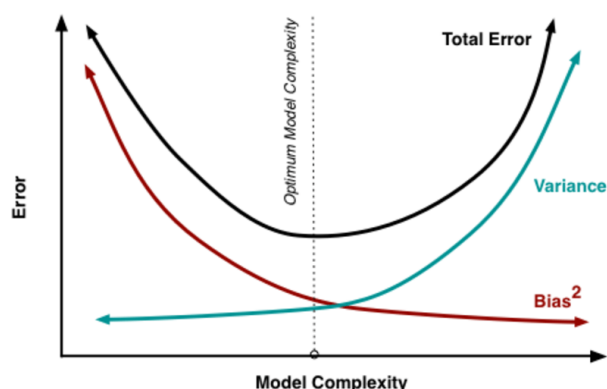
$$e = y - X\hat{\beta}.$$

Se desea, entonces, que tanto el sesgo como la varianza sean bajos, ya que los valores grandes dan como resultado predicciones deficientes del modelo. El estimador MCO al igual que el de máxima verosimilitud tiene la propiedad deseada de ser imparcial. Sin embargo, puede tener una gran variación la cual sucede cuando:

- Las variables predictoras están altamente correlacionadas entre sí;
- Hay muchos predictores. Esto se refleja en la fórmula para la varianza dada anteriormente: si m se aproxima a n , la varianza se acerca al infinito.

La solución general para esto es: reducir la varianza al costo de introducir cierto sesgo. Este enfoque se denomina regularización y casi siempre es beneficioso para el rendimiento predictivo del modelo. Como ejemplo tomemos el siguiente grafico:

Gráfico 1. Sesgo y variancia contra la complejidad del modelo.



Fuente: researchgate.net

Se observa que a medida que aumenta la complejidad del modelo, que en el caso de la regresión lineal puede considerarse como el número de predictores, la variancia de las estimaciones también aumenta, pero el sesgo disminuye. El MCO imparcial nos colocaría en el lado derecho de la imagen, que está lejos de ser óptimo. Es por eso que se regulariza: para reducir la variación a costa de algunos sesgos, moviéndonos hacia la izquierda en el gráfico, hacia el óptimo.

Tibshirani propuso un nuevo método llamado LASSO (Least Absolute Shrinkage and Selection Operator), el cual es una técnica de regresión lineal regularizada que combina un modelo de regresión con un procedimiento de contracción de algunos parámetros hacia cero y selección de variables, imponiendo una penalización sobre los coeficientes de regresión para mejorar la exactitud e interpretabilidad del modelo estadístico producido por este.

Tibshirani (1996) en su artículo propone el modelo minimizando la suma residual de cuadrados sujeto a que la suma del valor absoluto de los coeficientes sea menor que una constante, dicha restricción, produce coeficientes que tienden a 0 y, por lo tanto, da modelos interpretables.

En este método el parámetro de penalización λ se elige mediante validación cruzada, donde se selecciona un conjunto de valores para λ , se calcula el error de validación para cada valor y se elige el valor de λ con menor error, finalmente se reajusta el modelo con el valor de λ escogido.

El objetivo que se pretende abarcar en el presente estudio es realizar una estimación del modelo de regresión lasso con el método de máxima verosimilitud.

A continuación, se muestra la sección del modelo en la cual se describe la función de verosimilitud y propiedades del modelo como la potencia de la prueba, estimación de coeficientes, los residuos de deviancia y el umbral del parámetro λ . Además, se describen los resultados de la aplicación del modelo a un ejemplo empírico donde se evaluó la selección de variables y medidas de clasificación comparando los modelos con la regresión lasso y regresión logística. Finalmente se presenta una sección de conclusiones.

2. MODELO

2.1. Función de verosimilitud

Función de densidad de Y_i es $f_i(g(x_i^T \beta), y_i)$, donde g es una función de enlace conocida.

Se denota la función condicional de log-verosimilitud de Y_i por $l_i = \log f_i$.

La función de log verosimilitud penalizada es:

$$-\sum_{i=1}^n l_i(g(x_i^T \beta), y_i) + n \sum_{j=1}^d p_\lambda(|\beta_j|)$$

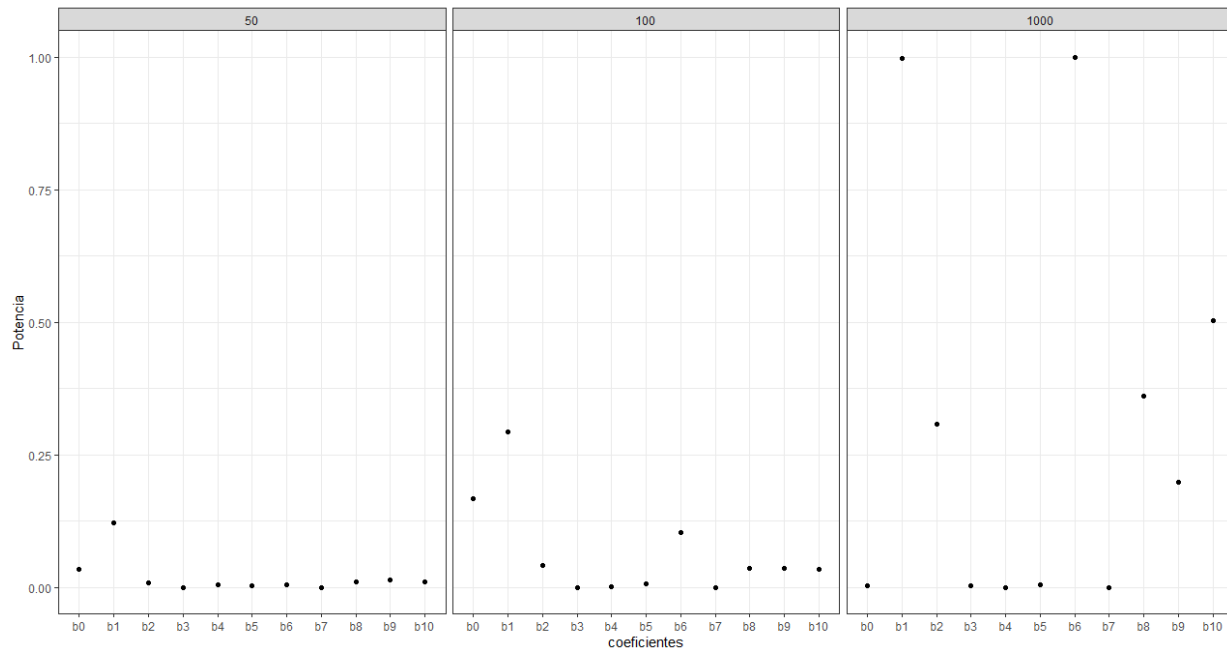
Para obtener un estimador de máxima verosimilitud penalizado de β se minimiza la función anterior respecto a β para algún umbral del parámetro λ .

2.2. Propiedades del modelo

Para evaluar las propiedades del modelo se utilizó una función de enlace logito en la función de log verosimilitud.

Se evaluó la potencia de la prueba de los coeficientes con simulaciones, se consideraron tres tamaños de muestra 50, 100 y 1000, con mil simulaciones se obtuvo la proporción de veces en las que cada uno de los once coeficientes son significativos. En el gráfico 2 se puede observar que a mayor tamaño de muestra la proporción de coeficientes significativos aumenta y la cantidad de coeficientes significativos es menor en tamaños de muestra pequeños que en tamaños grandes.

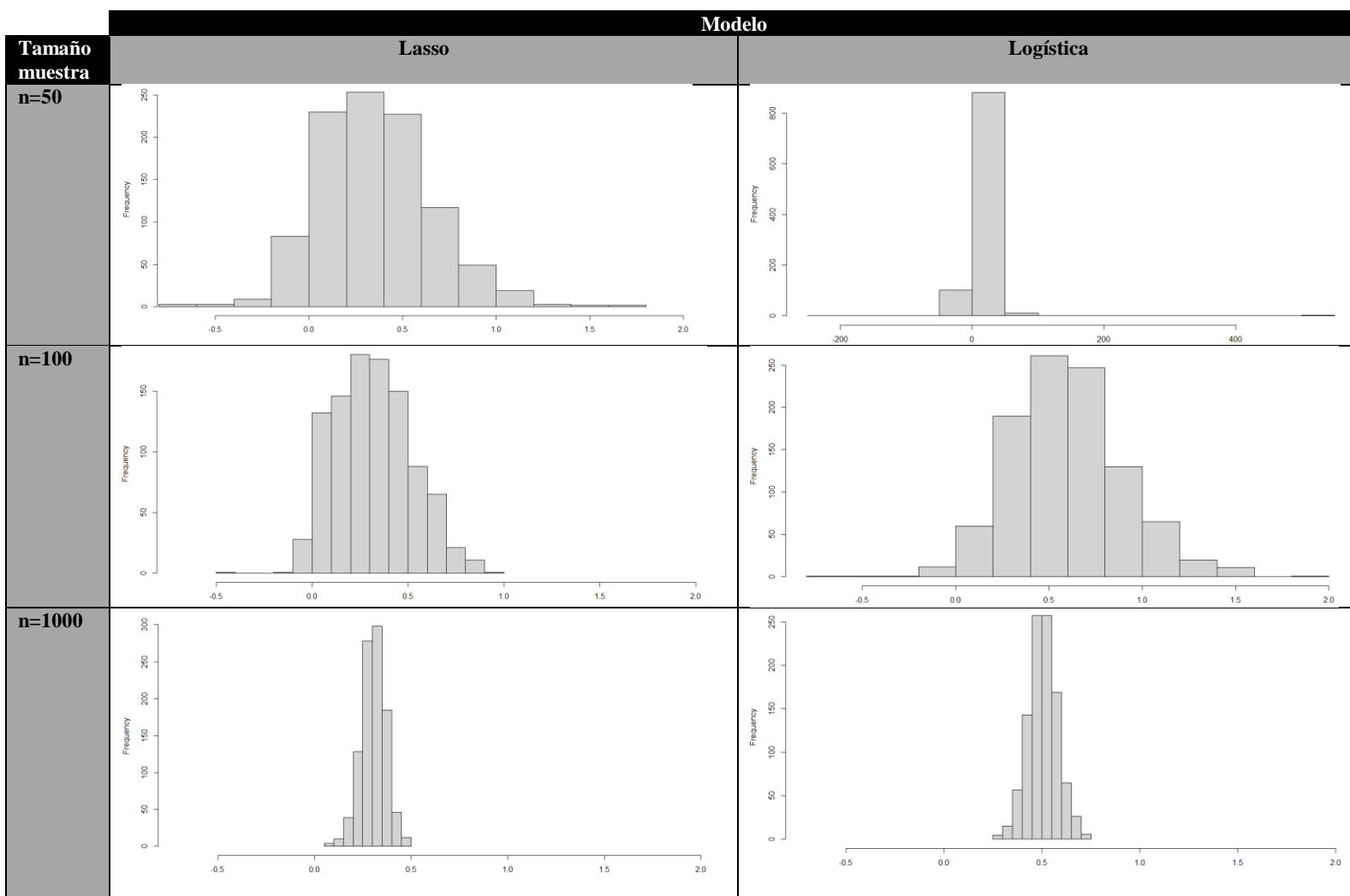
Gráfico 2. Potencia de la prueba para los coeficientes variando el tamaño de la muestra y con mil simulaciones



En la estimación de los coeficientes se consideraron tres tamaños de muestra 50, 100 y 1000, con mil simulaciones se obtuvo la estimación de los coeficientes para un modelo lasso y un modelo logístico, comparando las estimaciones obtenidas con un histograma. Se consideraron dos casos, uno en el que el coeficiente es significativo, b_1 , y otro en el que no lo es, b_4 .

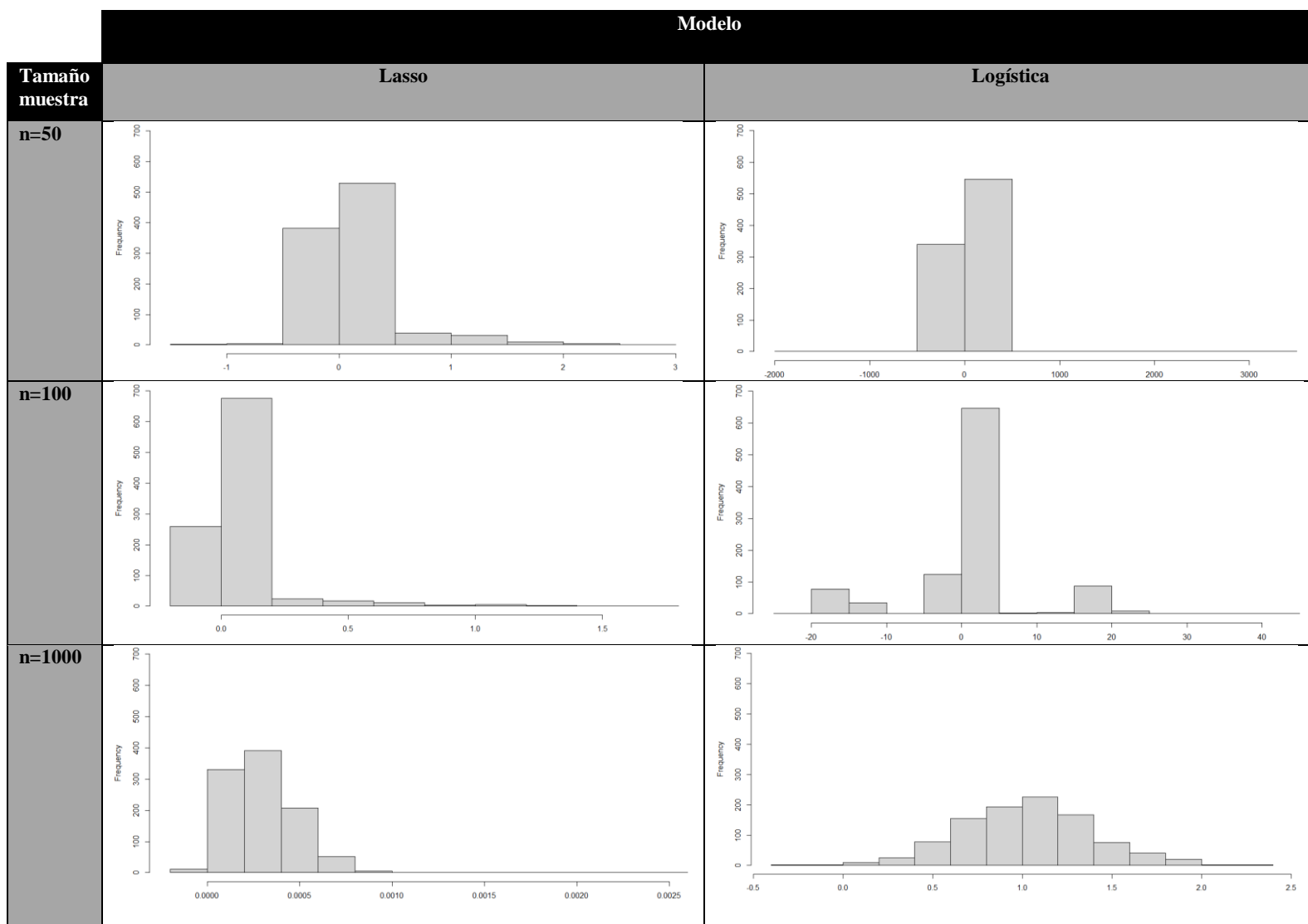
En el gráfico 3, se muestran los resultados para el caso en el que el coeficiente es significativo. Se puede observar que, a mayor tamaño de muestra, las estimaciones del coeficiente son más pequeñas en ambos modelos. También se muestra que las estimaciones en la regresión logística tienden a ser mayores que en la regresión lasso.

Gráfico 3. Histograma de la estimación del coeficiente significativo (b_1) variando el tamaño de muestra con mil simulaciones



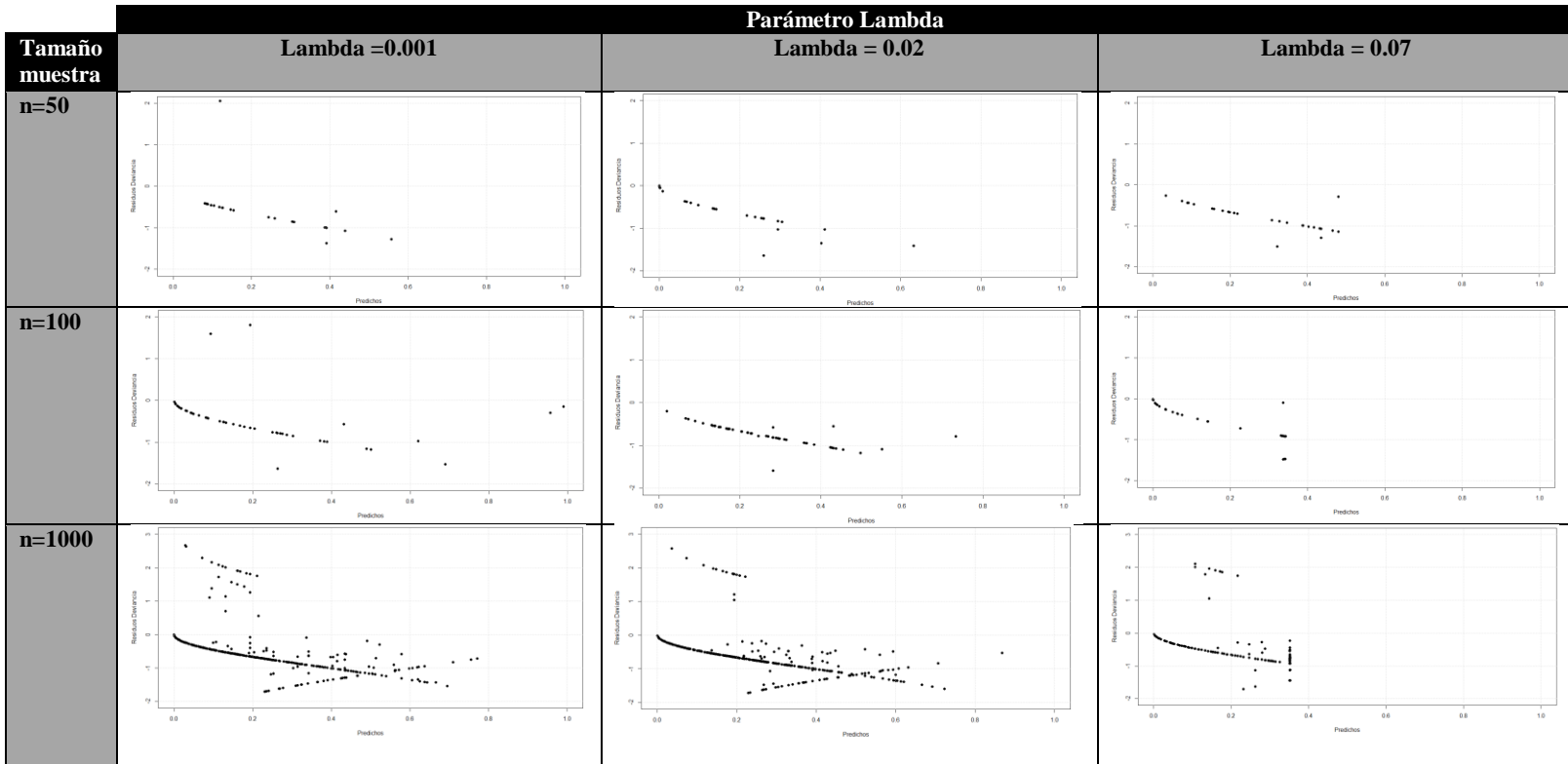
En el gráfico 4, se muestran los resultados para el caso del coeficiente no significativo. Se puede observar que, para la regresión lasso a mayor tamaño de muestra las estimaciones del coeficiente tienden a valores muy pequeños cercanos a cero. Mientras que para la regresión logística se observa que a mayor tamaño de muestra las estimaciones del coeficiente son más pequeñas, pero no tienden a valores cercanos a cero.

Gráfico 4. Histograma de la estimación del coeficiente no significativo (b4) variando el tamaño de muestra con mil simulaciones



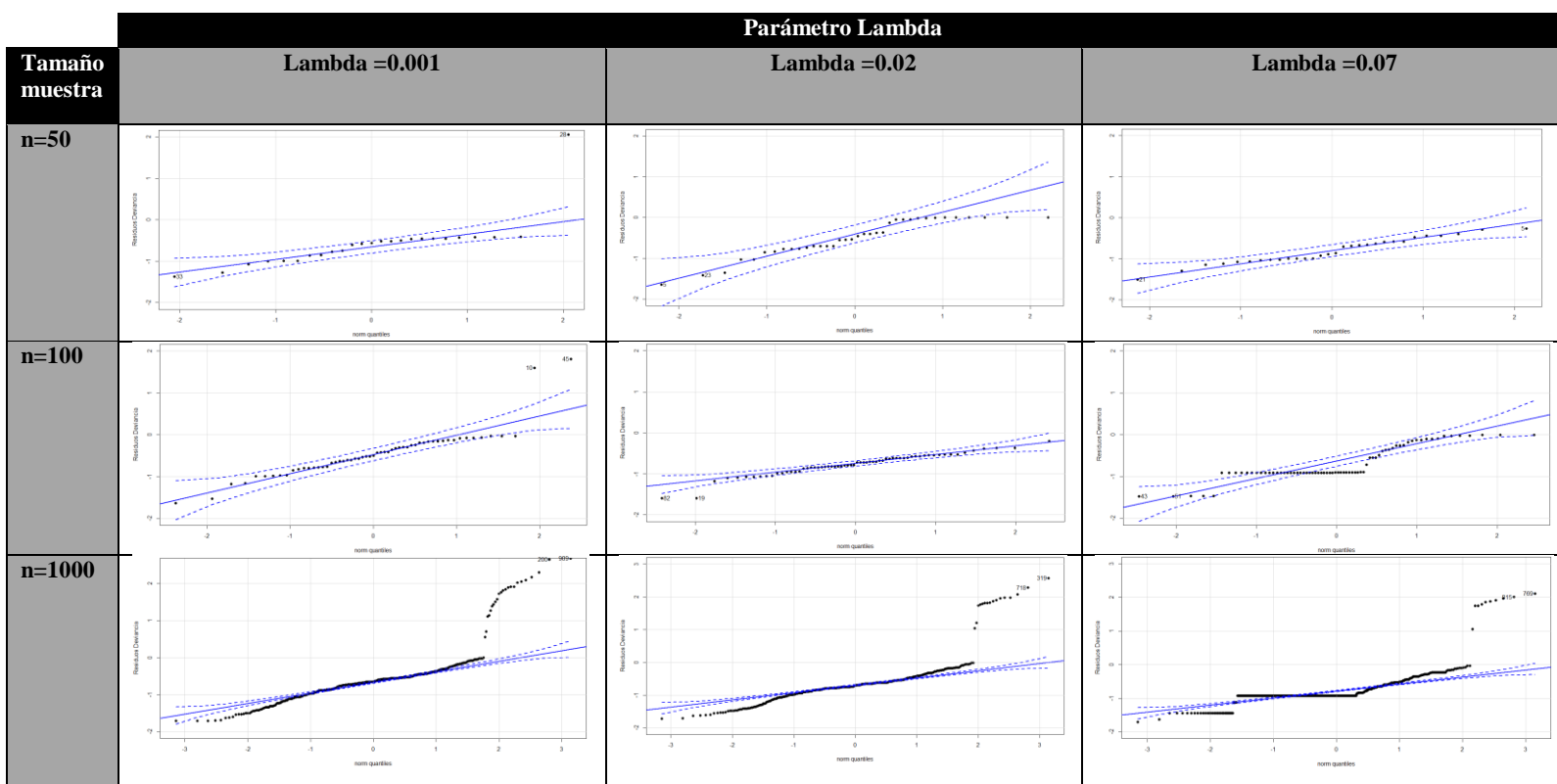
En el gráfico 5 se obtienen los gráficos de residuos de deviancia contra los valores predichos para el modelo lasso, en tres escenarios variando el valor del parámetro lambda y el tamaño de muestra. En los gráficos para los posibles escenarios y tamaños de muestra no se observa un patrón aleatorio, se muestra una ligera tendencia negativa, conforme los valores predichos aumentan los residuos disminuyen. También se observa que a mayor valor de lambda los predichos muestran valores bajos.

Gráfico 5. Residuos de deviancia vs Predicho variando el tamaño de muestra y el valor de lambda



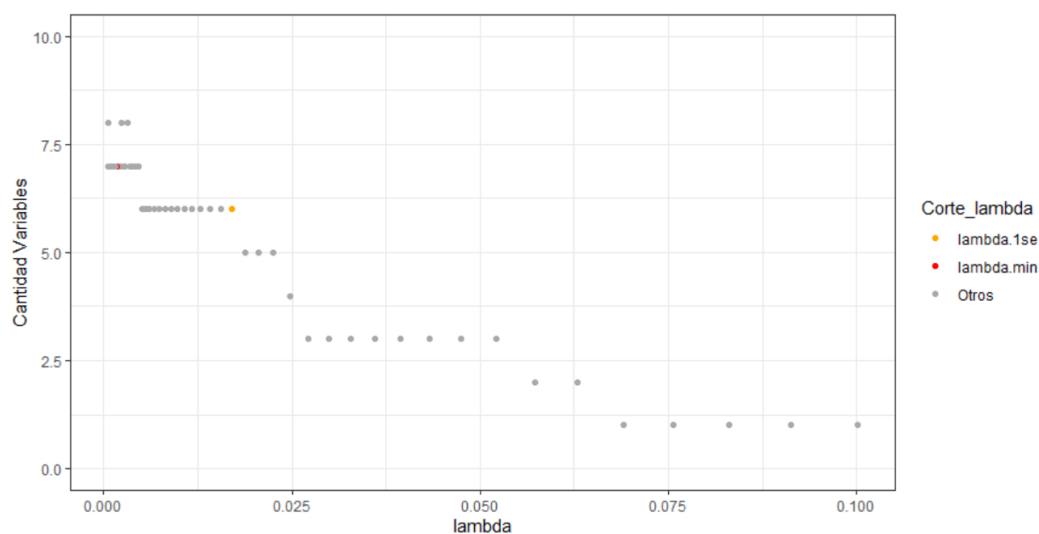
En el gráfico 6 se muestran los gráficos de normalidad qq plot donde se puede observar que, para las diferentes combinaciones de escenarios, los residuos de deviancia no se distribuyen como una normal. En tamaños de muestra altos se forma dos agrupaciones, una con residuos de deviancia positivos y altos, y otra con residuos de deviancia negativos.

Gráfico 6. Gráfico de normalidad QQ-Plot por cantidad de predictores y variando el tamaño de muestra



También se consideró el umbral lambda, donde se obtuvo la cantidad de variables significativas para distintos valores de lambda. En el gráfico 7 se observa que a mayor lambda menor es la cantidad de variables significativas. Se suele elegir los valores del umbral lambda.1se o el lambda.min pues el lambda.min es el valor de λ que da un error de validación cruzada medio mínimo. El otro λ es lambda.1se, que proporciona el modelo más regularizado de manera que el error está dentro de un error estándar del mínimo. (Hastie & Qian, 2014)

Gráfico 7. Cantidad de variables significativas según el lambda



2.3 Ejemplo empírico

Se utilizaron datos recolectados por el Banco Interamericano de Desarrollo (BID) que contiene variables sobre 9558 personas de 2973 hogares para Costa Rica. La variable objetivo se refiere a la categorización usando el PMT que clasifica a los hogares en “No vulnerables”, “Vulnerables”, “Pobres” y “En extrema pobreza”, en este caso se categorizó como 1 hogares pobres o en extrema pobreza y como 0 las categorías restantes.

Las variables utilizadas fueron:

- ninno: cantidad de niños.
- cocina_elec: cocina con electricidad.
- banno: tenencia de baño.
- sin_educ: jefe del hogar sin educación.
- urbano: zona urbana.
- renta: monto de pago de alquiler en colones.
- sani_alcant: sanitario por alcantarillado.
- casadx: jefe del hogar casado.
- dependencia: miembros del hogar menores de 19 años o mayores de 64.
- casa_propia: si la casa es propia.

Además, se realizó una comparación de los resultados con la regresión logística.

Al comparar los coeficientes significativos de los dos modelos se observa que los obtenidos por la regresión lasso también son significativos en el modelo de la regresión logística y las estimaciones de dichos coeficientes son similares, así como los respectivos errores estándar, sin embargo, las estimaciones de la regresión lasso son menores. En este caso la regresión logística presenta una mayor cantidad de variables significativas al compararla con la regresión lasso.

Cuadro 1. Variables significativas con regresión lasso y logística

Variable	Regresión Lasso			Regresión Logística		
	Coefficiente	Error Estándar	Valor P	Coefficiente	Error Estándar	Valor P
intercepto	-0.0009	0.5219	0.9986	0.3058	0.543	0.5733
ninno	0.3648	0.0378	< 0.0001 *	0.4967	0.0405	< 0.0001 *
cocina_elec	-0.3363	0.0954	0.0004 *	-0.4657	0.1007	< 0.0001 *
banno	-0.7896	0.5113	0.1225	-1.2839	0.532	0.0158 *
sin_educ	0.00003	0.2115	0.9999	0.9898	0.1996	< 0.0001 *
urbano	-0.0008	0.1037	0.994	0.0052	0.1083	0.9619
renta	-0.0098	0.0012	< 0.0001 *	-0.0094	0.0013	< 0.0001 *
sani_alcant	-0.0005	0.1187	9.96E-01	-0.1927	0.1284	0.1332
casadx	-0.3456	0.0953	0.0003 *	-0.5526	0.1016	< 0.0001 *
dependencia	0.0545	0.0208	0.0087 *	0.0976	0.0214	< 0.0001 *
casa_propia	-0.4381	0.1204	0.0003 *	-0.4647	0.1249	0.0002 *

En cuanto al error cuadrático medio se obtuvo que el modelo de regresión logística es el que tiene un valor menor de 0.30.

Cuadro 2. Error cuadrático medio para la regresión lasso y la regresión logística

Modelo	MSE
Logística	0.302
Lasso	0.427

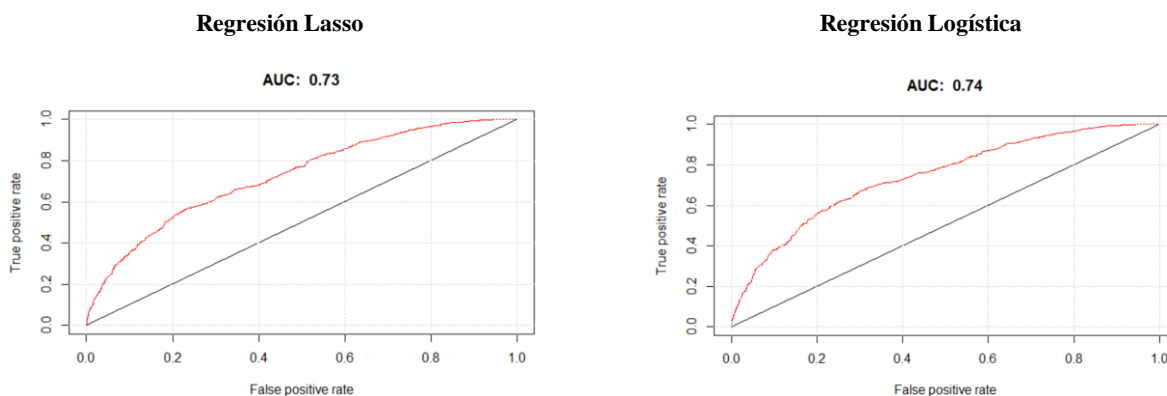
Al evaluar las medidas de precisión para la clasificación se utilizó como punto de corte la media de la variable respuesta que corresponde a 0.22. Se obtuvo que el modelo de regresión lasso tiene una sensibilidad de 76% mientras que el modelo de la regresión logística tiene un 65%.

Cuadro 3. Medidas de precisión de los modelos de regresión lasso y regresión logística

Modelo	Sensibilidad	Especificidad
Logística	0.655	0.711
Lasso	0.761	0.518

En el gráfico 8, se muestran las curvas ROC para ambos modelos, en los cuales se obtuvo un área bajo la curva similar de 0.73.

Gráfico 8. Curva ROC para la regresión lasso y la regresión logística



3. CONCLUSIONES

El modelo de regresión lasso permite realizar selección de variables cuando se tiene una gran cantidad de variables, en este caso se comprobó que las estimaciones de los coeficientes son más precisas con la regresión lasso que las obtenidas con el modelo de regresión logística.

Se observó que el parámetro lambda usado es de gran importancia, ya que influye en las predicciones y en la cantidad de variables significativas.

En el ejemplo utilizado acerca de la pobreza se obtuvo que el modelo de regresión lasso da una mejor sensibilidad para la categoría de hogares pobres que el modelo logístico. En cuanto a la selección de variables con la regresión lasso se obtuvo que las variables se redujeron de 10 a 6, resultando significativas las variables de: cantidad de niños, si en el hogar se cocina con electricidad, el pago de alquiler, si el jefe está casado, dependencia y la tenencia de casa propia.

Destacan variables de composición del hogar como la cantidad de niños y la relación de dependencia que evidencian que familias con más niños son más propensos a caer en pobreza. En relación a los jefes de hogar se encontró que el estado civil de estar casado disminuye la propensión de caer en pobreza. Así mismo, la tenencia de casa propia disminuye la propensión de caer en pobreza.

4. BIBLIOGRAFÍA

Fan, J., Li, R. (1999). Variable Selection via Penalized Likelihood. California Digital Library: Universidad de California.

Hastie, T., & Qian, J. (26 de Junio de 2014). Lasso and Elastic-Net Regularized Generalized Linear Models. Obtenido de https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. Journal of Royal Statistical Society, B, 58, 267-288.

5. ANEXO

Código de R

```
library(glmnet)
library(dplyr)
library(tidyr)
library(stats4)
library(bbmle)
library(ggplot2)
library(car)

load("hogar.Rdata")

LL <- function(b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10) {
  n<-nrow(hogar)
  x<-model.matrix(pobre~.,hogar)

  betas = c(b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10)
  xb = x%*%betas
  xb_g = 1/(1+exp(-xb))
  vero1 = suppressWarnings(dbinom(hogar$pobre,
                                   size = 1,
                                   prob = xb_g,
                                   log=FALSE))
  parte1<- -sum(log(vero1))
  parte2<-sum(abs(betas))

  vero= parte1+(n*lambda*parte2)
}
```

```
# Usando validacion cruzada para determinar el mejor Lambda
x<-model.matrix(pobre~.,hogar)
sal.cv<-cv.glmnet(x[,-1],hogar$pobre,alpha=1, family = binomial)
#El lambda de 1 error estandar
lambda<-sal.cv$lambda.1se
```

```
## Valores iniciales a partir de la salida de glmnet
```

```
mod1 <- mle(LL, start = list(  b0= -1,
                                b1= 0.38,
                                b2= -0.32,
                                b3= -0.32,
                                b4= 0.65,
                                b5= 0.0000000001,
                                b6= -0.005,
                                b7= 0.0000000001,
                                b8= -0.35,
                                b9= 0.05,
                                b10= 0.0000000001
                                ))
summary(mod1)
```

```
mod2 <- mle2(LL, start = list(
  b0= -1,
  b1= 0.38,
  b2= -0.32,
  b3= -0.32,
  b4= 0.65,
  b5= 0.0000000001,
  b6= -0.005,
```

```

b7= 0.0000000001,
b8= -0.35,
b9= 0.05,
b10= 0.0000000001
))

```

```

summary(mod2)
colnames(hogar)[-1]

```

```

# Mean square error
MSE <- function(Pred,Real) {
  N<-length(Real)
  ss<-(1/N)*sum((Real-Pred)^2)
  return(ss)
}

```

```

#lasso
x<-model.matrix(pobre~.,hogar)
coef_las<-summary(mod2)@coef[,1]

pred_las<-1/(1+exp(-(x %*% coef_las)))
pred_las_class<-ifelse(pred_las>mean(hogar$pobre),1,0)
MSE(pred_las_class,hogar$pobre)

```

```

# sensibilidad
MC1<-table(hogar$pobre,pred_las_class)
MC1

```

```
pp1 <- round(diag(MC1)[2]/rowSums(MC1)[2],4)
```

```
pp1
```

```
pn1<- round(diag(MC1)[1]/rowSums(MC1)[1],4)
```

```
pn1
```

```
#curva ROC
```

```
pred <- ROCR::prediction(pred_las,hogar$pobre )
```

```
auc <- ROCR::performance(pred, "auc")
```

```
auc <- attributes(auc)$y.values[[1]]
```

```
perf <- ROCR::performance(pred,"tpr","fpr")
```

```
plot(perf,col="red",main=paste("AUC: ",round(auc,2)))
```

```
segments(0,0,1,1,col='black')
```

```
grid()
```

```
#regresion logistica
```

```
mod_log <- glm(pobre~., family = binomial(link = "logit"), data=hogar)
```

```
summary(mod_log)
```

```
pred_log<-predict(mod_log,hogar[,-1],type="response")
```

```
pred_log_class<-ifelse(pred_log>mean(hogar$pobre),1,0)
```

```
MSE(pred_log_class,hogar$pobre)
```

```
# sensibilidad
```

```
MC<-table(hogar$pobre,pred_log_class)
```

```
MC
```

```
pp <- round(diag(MC)[2]/rowSums(MC)[2],4)
```

```
pp
```

```
pn<- round(diag(MC)[1]/rowSums(MC)[1],4)
```

```
pn
```

```
#curva ROC
```

```
pred <- ROCR::prediction(pred_log,hogar$pobre )
```

```
auc <- ROCR::performance(pred, "auc")
```

```
auc <- attributes(auc)$y.values[[1]]
```

```
perf <- ROCR::performance(pred,"tpr","fpr")
```

```
plot(perf,col="red",main=paste("AUC: ",round(auc,2)))
```

```
segments(0,0,1,1,col='black')
```

```
grid()
```

```
#####Cantidad de variables seleccionadas según el lambda
```

```
# Usando validacion cruzada para determinar el mejor Lambda
```

```
n<-nrow(hogar)
```

```
x<-model.matrix(pobre~.,hogar)
```

```

sal.cv<-cv.glmnet(x[,-1],hogar$pobre,alpha=1, family = binomial)
#El lambda de 1 error estandar
lambda_tot<-sal.cv[["lambda"]]

mat1<-matrix(rep(0,length(lambda_tot)*1),
              nrow=length(lambda_tot),ncol = 1,byrow = T)

for (i in 1:length(lambda_tot)) {

  lambda<-lambda_tot[i]

  mod2 <- mle2(LL, start = list(
    b0= -1,
    b1= 0.38,
    b2= -0.32,
    b3= -0.32,
    b4= 0.65,
    b5= 0.0000000001,
    b6= -0.005,
    b7= 0.0000000001,
    b8= -0.35,
    b9= 0.05,
    b10= 0.0000000001
  ))

  smod<-summary(mod2)
  mat1[i,]<-sum(1*(smod@coef[,4]<0.05))

}

```



```

df4<-data.frame(lambda=lambda_tot,
                 Cant_Variables=mat1[,1])

df4<-df4 %>%
  mutate(Corte_lambda=ifelse(lambda==sal.cv$lambda.1se,"lambda.1se",ifelse(
    lambda==sal.cv$lambda.min,"lambda.min","Otros" )))

ggplot(data = df4,aes(lambda,Cant_Variables)) +
  geom_point(aes(colour=Corte_lambda))+
  ylim(0,10)+
  ylab("Cantidad Variables")+
  theme_bw()+
  scale_color_manual(values=c("orange", "red", "dark gray"))

##### simulaciones #####

#1. Potencia de la prueba de los coeficientes

x<-model.matrix(pobre~.,hogar)
sal.cv<-cv.glmnet(x[,-1],hogar$pobre,alpha=1, family = binomial)
#El lambda de 1 error estandar
lambda<-sal.cv$lambda.1se

muestras<-c(50,100,1000)

```

```
#matrices que guardan los resultados de la potencia
```

```
tam1<-matrix(rep(0,11*1000),nrow=1000,ncol = 11,byrow = T)
```

```
tam2<-matrix(rep(0,11*1000),nrow=1000,ncol = 11,byrow = T)
```

```
tam3<-matrix(rep(0,11*1000),nrow=1000,ncol = 11,byrow = T)
```

```
#matrices que guardan los resultados de las estimaciones lasso
```

```
tam1_es<-matrix(rep(0,11*1000),nrow=1000,ncol = 11,byrow = T)
```

```
tam2_es<-matrix(rep(0,11*1000),nrow=1000,ncol = 11,byrow = T)
```

```
tam3_es<-matrix(rep(0,11*1000),nrow=1000,ncol = 11,byrow = T)
```

```
#matrices que guardan los resultados de las estimaciones logistica
```

```
tam1_es_log<-matrix(rep(0,11*1000),nrow=1000,ncol = 11,byrow = T)
```

```
tam2_es_log<-matrix(rep(0,11*1000),nrow=1000,ncol = 11,byrow = T)
```

```
tam3_es_log<-matrix(rep(0,11*1000),nrow=1000,ncol = 11,byrow = T)
```

```
for(i in 1:length(muestras)){
```

```
  for (j in 1:1000) {
```

```
    #muestra con reemplazo
```

```
    index<-sample(1:nrow(hogar),muestras[i], replace=TRUE)
```

```
    hogar_temp<- hogar[index,]
```

```
    n<-nrow(hogar_temp)
```

```
    x<-model.matrix(pobre~.,hogar_temp)
```

```
LL_temp <- function(b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10) {
```

```
  betas = c(b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10)
```

```
  xb = x%*%betas
```

```
  xb_g = 1/(1+exp(-xb))
```

```

vero1 = suppressWarnings(dbinom(hogar_temp$pobre,
                                size = 1,
                                prob = xb_g,
                                log=FALSE))
parte1<- -sum(log(vero1))
parte2<-sum(abs(betas))

vero= parte1+(n*lambda*parte2)
}

mod2 <- mle2(LL_temp, start = list(
  b0= -1,
  b1= 0.38,
  b2= -0.32,
  b3= -0.32,
  b4= 0.65,
  b5= 0.0000000001,
  b6= -0.005,
  b7= 0.0000000001,
  b8= -0.35,
  b9= 0.05,
  b10= 0.0000000001
))

smod<-summary(mod2)

mod_log <- glm(pobre~., family = binomial(link = "logit"), data=hogar_temp)

# extraer valores

```

```

if(i==1){
  tam1[j,]<- 1*(smod@coef[,4]<0.05)
  tam1_es[j,]<- smod@coef[,1]
  tam1_es_log[j,]<- coef(mod_log)
}

if(i==2){
  tam2[j,]<- 1*(smod@coef[,4]<0.05)
  tam2_es[j,]<- smod@coef[,1]
  tam2_es_log[j,]<- coef(mod_log)
}

if(i==3){
  tam3[j,]<- 1*(smod@coef[,4]<0.05)
  tam3_es[j,]<- smod@coef[,1]
  tam3_es_log[j,]<- coef(mod_log)
}
}
}

#resumen para potencia
mean1<-apply(tam1,2,function(x) mean(x,na.rm = TRUE))
mean2<-apply(tam2,2,function(x) mean(x,na.rm = TRUE))
mean3<-apply(tam3,2,function(x) mean(x,na.rm = TRUE))

coeficientes<-c("b0","b1","b2","b3","b4","b5","b6","b7","b8","b9","b10")
df<-data.frame(coeficientes=c(coeficientes,coeficientes,coeficientes),

Muestra=c(rep(muestras[1],length(coeficientes)),rep(muestras[2],length(coeficientes)),rep(muestras[3],length(coeficientes))),

```

```

Potencia=c(mean1,mean2,mean3))
#grafico potencia
df$coeficientes<-factor(df$coeficientes,
c("b0","b1","b2","b3","b4","b5","b6","b7","b8","b9","b10"))

```

levels =

```

ggplot(data = df,aes(coeficientes,Potencia)) +
  geom_point()+
  ylim(0,1)+
  theme_bw() +
  facet_wrap(~Muestra, ncol =3)

```

#2. Histogramas de estimacion de coeficientes

```
#coef significativo
```

```
#n=50
```

```
#lasso
```

```
hist(tam1_es[,2],xlab = "",xlim = c(-0.7,2))
```

```
hist(tam1_es_log[-676,2],xlab = "")
```

```
#n=100
```

```
#lasso
```

```
hist(tam2_es[,2],xlab = "",xlim = c(-0.7,2))
```

```
hist(tam2_es_log[,2],xlab = "")
```

```
#n=1000
```

```
#lasso
```

```
hist(tam3_es[,2],xlab = "",xlim = c(-0.7,2))
```

```
hist(tam3_es_log[,2],xlab = "",xlim = c(-0.7,2))
```

```
#coef no significativo
```

```
#n=50
```

```
#lasso
```

```
hist(tam1_es[,5],xlab = "",ylim = c(0,700))
```

```
hist(tam1_es_log[-c(676),5],xlab = "",ylim = c(0,700))
```

```
#n=100
```

```
#lasso
```

```
hist(tam2_es[,5],xlab = "",ylim = c(0,700))
```

```
hist(tam2_es_log[,5],xlab = "",ylim = c(0,700))
```

```
#n=1000
```

```
#lasso
```

```
hist(tam3_es[,5],xlab = "",ylim = c(0,700))
```

```
hist(tam3_es_log[,5],xlab = "",ylim = c(0,700))
```

#3. Análisis de residuos de deviancia (linealidad y normalidad)

```
x<-model.matrix(pobre~.,hogar)
sal.cv<-cv.glmnet(x[,-1],hogar$pobre,alpha=1, family = binomial)
#lambda<-sal.cv$lambda.1se
#lambda<-sal.cv$lambda.min
lambda<-0.07
```

```
#50 , 100, 1000
```

```
muestras<-1000
```

```
#muestra con reemplazo
```

```
index<-sample(1:nrow(hogar),muestras, replace=TRUE)
```

```
hogar_temp<- hogar[index,]
```

```
n<-nrow(hogar_temp)
```

```
x<-model.matrix(pobre~.,hogar_temp)
```

```
LL_temp <- function(b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10) {
```

```
  betas = c(b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10)
```

```
  xb = x%*%betas
```

```
  xb_g = 1/(1+exp(-xb))
```

```
  vero1 = suppressWarnings(dbinom(hogar_temp$pobre,
```

```
    size = 1,
```

```
    prob = xb_g,
```

```
    log=FALSE))
```

```
  parte1<- -sum(log(vero1))
```

```
  parte2<-sum(abs(betas))
```

```

    vero= parte1+(n*lambda*parte2)
  }

```

```

mod2 <- mle2(LL_temp, start = list(
  b0= -1,
  b1= 0.38,
  b2= -0.32,
  b3= -0.32,
  b4= 0.65,
  b5= 0.0000000001,
  b6= -0.005,
  b7= 0.0000000001,
  b8= -0.35,
  b9= 0.05,
  b10= 0.0000000001
))

```

```

smod<-summary(mod2)
coef_las<-smod@coef[,1]
pred_las<-1/(1+exp(-(x %*% coef_las)))
pred_las_class<-ifelse(pred_las>mean(hogar$pobre),1,0)

```

```

res<-hogar_temp$pobre-pred_las_class
signo.res=recode(res,"lo:0=-1;0:hi=1") #extrae el signo

```

```

#logverosimilitud

```

```

#modelo observado

```



```

vero1 = suppressWarnings(dbinom(hogar_temp$pobre,
                                size = 1,
                                prob = pred_las,
                                log=FALSE))
parte1<-log(vero1)
parte2<-sum(abs(coef_las))
log.veros.observado= parte1+(n*lambda*parte2)

#modelo saturado

#Patrones de covariancia
data_tm<-hogar_temp %>% group_by(ninnos,cocina_elec,banno,sin_educ,
                                urbano,renta,sani_alcant,casadx,dependencia,
                                casa_propia) %>%

summarise(
  y=sum(pobre),
  ene=n()
)

prob<-data_tm$y/data_tm$ene

vero1_2 = suppressWarnings(dbinom(hogar_temp$pobre,
                                size = 1,
                                prob = prob,
                                log=FALSE))
parte1_2<-log(vero1_2)
log.veros.saturado= parte1_2+(n*lambda*parte2)

```

```
#residuos de deviancia
res.dev=signo.res*(2*(log.veros.saturado-log.veros.observado))^.5

plot(res.dev~pred_las,xlab="Predichos",ylab = "Residuos Deviancia",main="",
      pch=19,xlim=c(0,1),ylim=c(-2,3))
grid()

qqPlot(res.dev,ylab = "Residuos Deviancia",pch=19,ylim = c(-2,3))
grid()
```